EEU4C21/CSP55031/EEP55C26:
Open Reconfigurable Networks

Building blocks of a wireless communication system - 1

Communication System

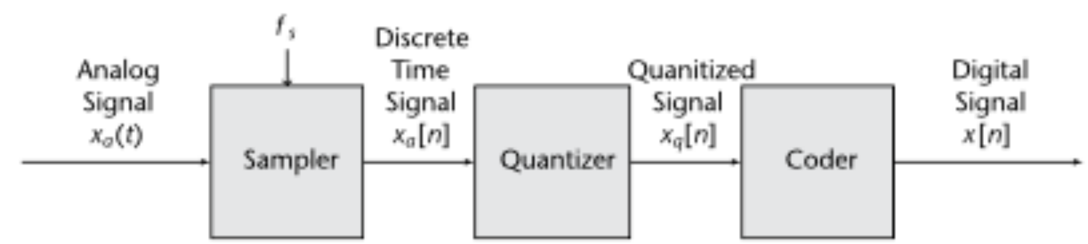
- Information in the form of a message sample, usually represented as ***m[n]***
    - Think of this as a array of binary vectors **m** with **n** as the sample index
- Transmitter converts it a continuous time domain RF signal - `s(t)`
- Channel introduces distortions to the transmitted information, typically in the form of noise
    - Characterised by the additive nature as a continuous time signal `n(t)`
- Receiver observers the distorted signal information `r(t)`
- Challenge is to recover the estimate of the original message sample ($\hat{m}[n]$) given that the observed signal was `r(t)`
- To make the system robust and resilient to channel conditions, signal processing techniques are employed at the transmitter and receiver ends

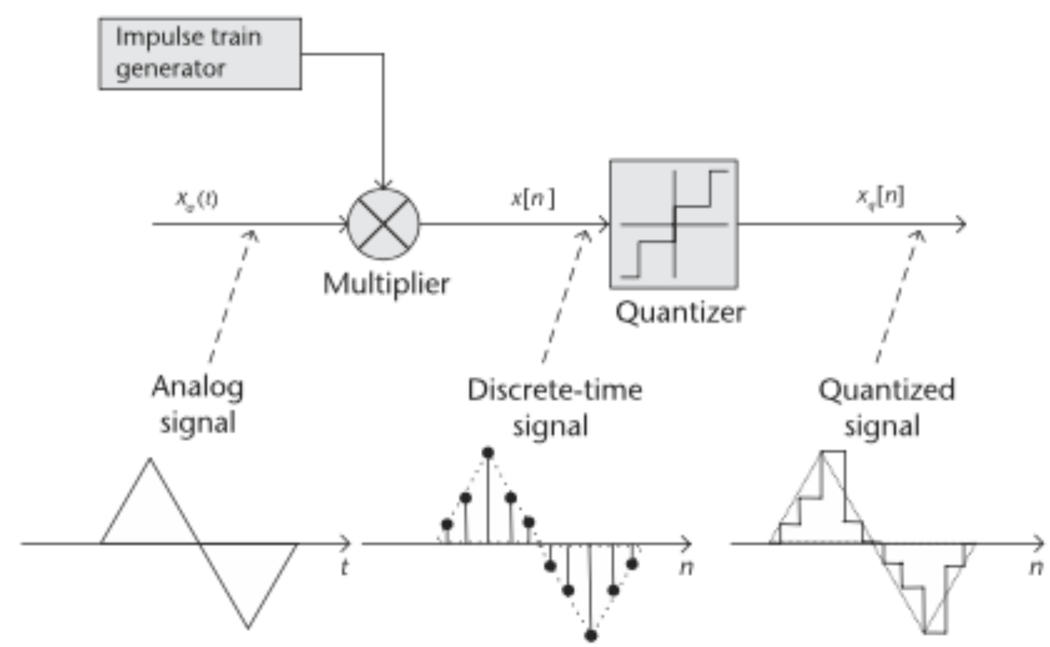
$$\text{Binary Information Source} \rightarrow m[n]$$
$$\downarrow$$
$$\text{Transmitter} \rightarrow s(t)$$
$$\downarrow$$
$$\text{Channel} \rightarrow s(t) + n(t)$$
$$\downarrow$$
$$\text{Receiver} \leftarrow r(t)$$
$$\downarrow$$
$$\text{Binary Information Sink} \leftarrow \hat{m}[n]$$

Digitising source info

Binary Information Source $\rightarrow m[n]$
$\downarrow$
Transmitter $\rightarrow s(t)$
$\downarrow$
Channel $\rightarrow s(t) + n(t)$
$\downarrow$
Receiver $\leftarrow r(t)$
$\downarrow$
Binary Information Sink $\leftarrow \hat{m}[n]$

- Information in real-world is typically an analogue signal
- A sampling block periodically observes the value of real world signal - generates a discrete-time observation
- Depending on the data-width, a quantisation approach is chosen to map input observations to finite data range
- Sampling rate chosen must satisfy the Nyquist criterion to ensure accurate reproduction of the real-world signal





Digitising source info

Binary Information Source $\rightarrow m[n]$
$\downarrow$
Transmitter $\rightarrow s(t)$
$\downarrow$
Channel $\rightarrow s(t) + n(t)$
$\downarrow$
Receiver $\leftarrow r(t)$
$\downarrow$
Binary Information Sink $\leftarrow \hat{m}[n]$

- Sampling rate chosen must satisfy the Nyquist criterion to ensure accurate reproduction of the real-world signal

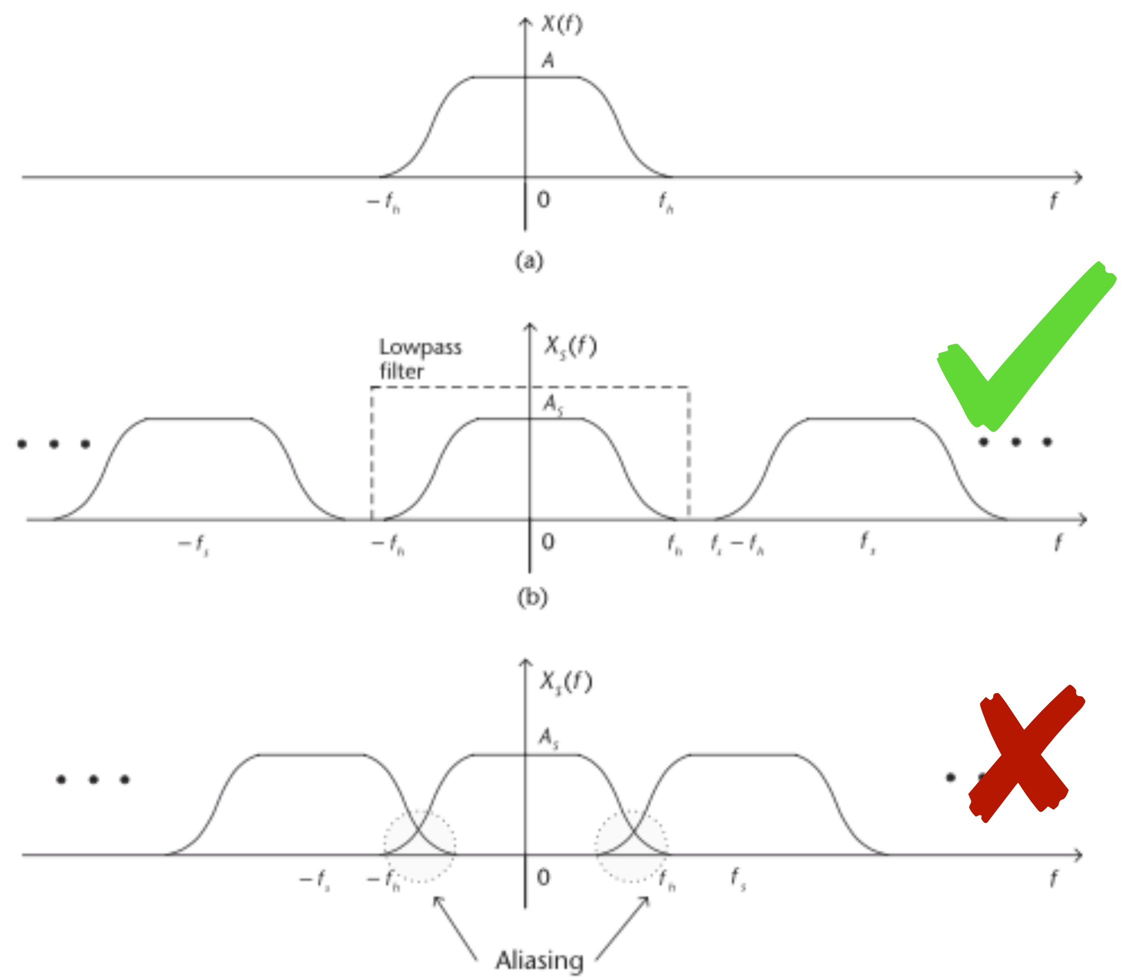
  - i.e.,

for $x(t = nT_s) \rightarrow x[n], -\infty < n < \infty, f_s = 1/T_s$

then $T_s \geq \dfrac{1}{2B}$ **for accurate reconstruction**

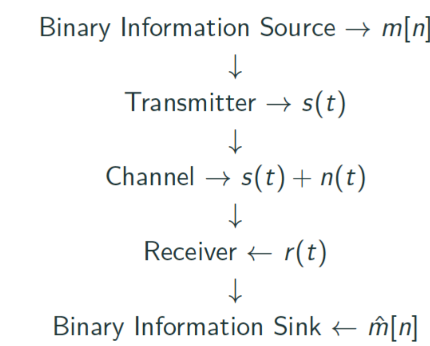
given $X(\omega) = 0, |\omega| \geq 2\pi B$

i.e., signal is bandlimited




```
   Fs = 1000;       % Sample rate (Hz)
   Fa = 1105;       % Input Frequency (Hz)
 % Determine Nyquist zones
   zone = 1 + floor(Fa / (Fs/2));
   alias = mod(Fa, Fs);
   if ~mod(zone,2) % 2nd, 4th, 6th, ... Nyquist Zone
     alias = -(Fs - alias)/Fs;
   else
     alias = (alias)/Fs;
   end
 % Create the analog/time domain and digital sampling vectors
   N = 2*1/abs(alias) + 1;          % Number of Digital samples
   points = 256;                    % Analog points between digital
samples
   analogIndexes = 0:1/points:N-1;
   samplingIndexes = 1:points:length(analogIndexes);
   wave = sin(2*pi*Fa/Fs*analogIndexes);
```

Digitising source info



- Sample rate conversion is employed to reduce data-rate (down-sample) or to compress the bandwidth (up-sample)
  - Down-sampling through <u>decimation</u> ignores every sample other than multiples of decimation factor `D`

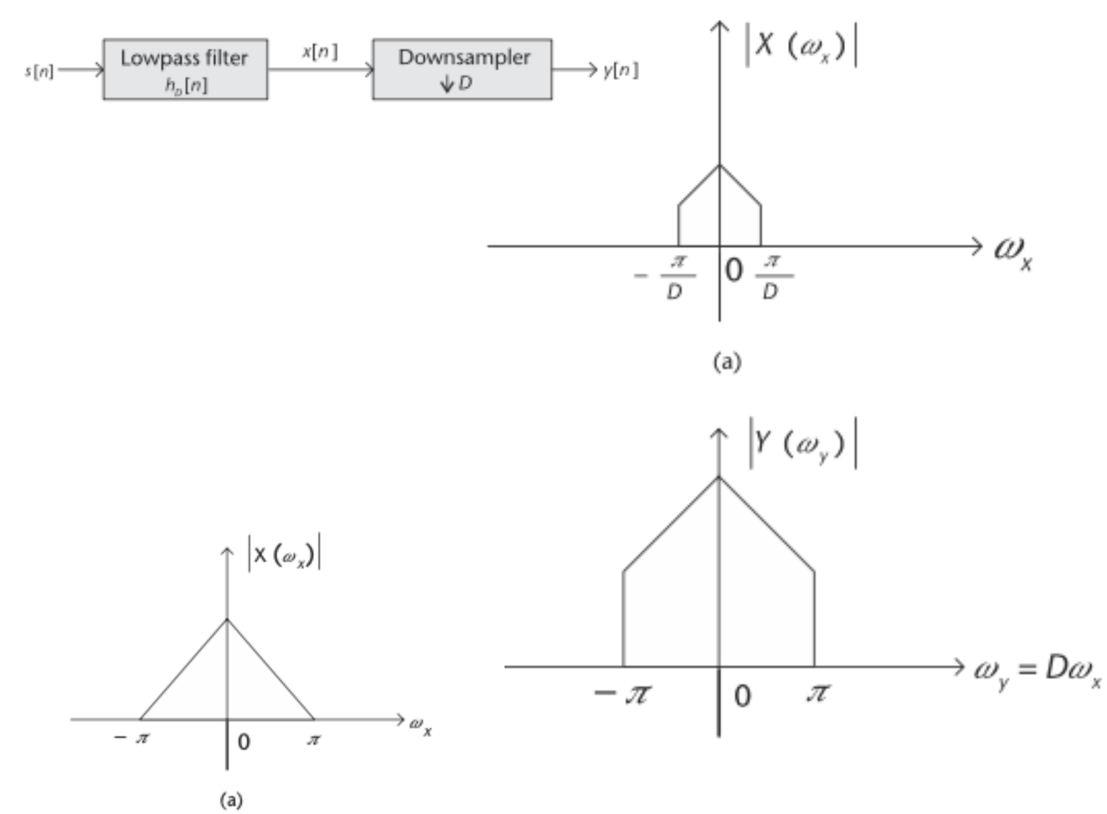
    i.e., $y[n] = x[nD], D = 1,2,3...$

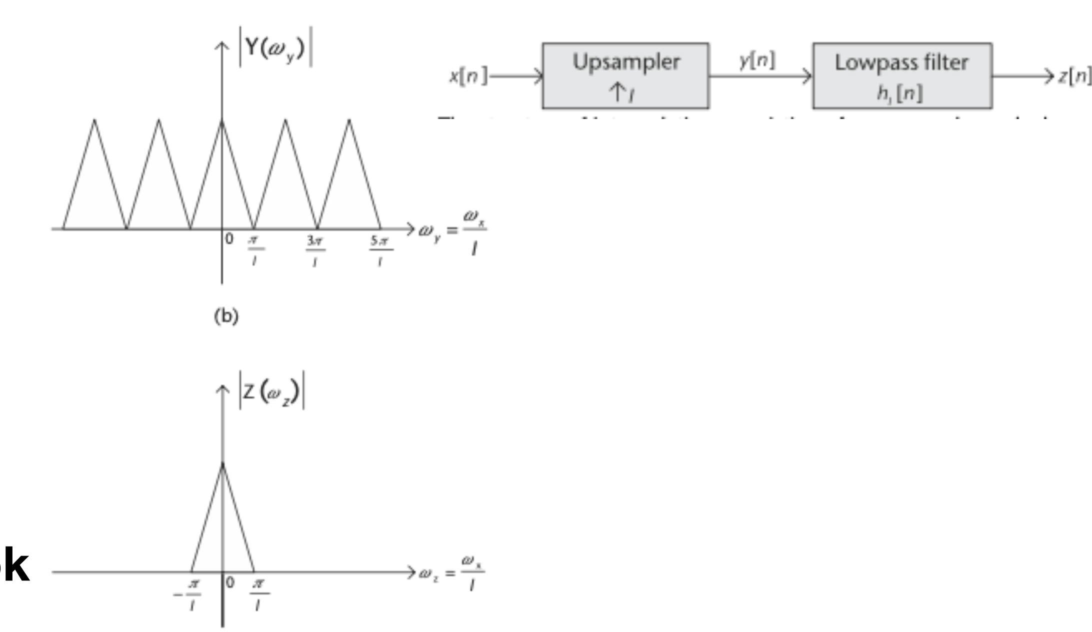
  - The frequency range of the resultant signal gets stretched by the same factor

    i.e., $\omega_y = D \times \omega_x$

    - Requires that $0 \leq |\omega_x| \leq \frac{\pi}{D}$

  - Up-conversion through <u>interpolation</u> injects 0's at `I-1` samples between each existing sample, where `I` is the interpolation factor
  - Compresses the frequency of the interpolated signal by a factor of `I`





Visualise this on MATLAB: see code 2.3 to 2.7 in the textbook

Signal representation

We usually represent signals using the In-phase and Quadrature-phase components

- Either in time or frequency domain $I(t), Q(t)$ or $I(\omega), Q(\omega)$
- Quadrature component can be thought of as imaginary part of a complex-valued signal

$$\implies s(t) = I(t) + jQ(t)$$

  - Euler's rules and other mathematical operators from complex numbers can be easily applied for many operations

For e.g., consider $I(t) = cos\,(\omega t)$ and $Q(t) = sin\,(\omega t)$

Then, a simple frequency shift [up-conversion] on `s(t)` can be expressed by $r(t) = s(t)e^{j\omega_c t}$

$$s(t)e^{j\omega_c t} = cos(\omega t)e^{j\omega_c t} + j\,sin(\omega t)e^{j\omega_c t}$$
$$= \left(cos\omega t \times cos\omega_c t - sin\omega t \ \times \sin\omega_c t)\right) + j\left(sin\omega t \times \cos\omega_c t + \cos\omega t \times \sin\omega_c t\right)$$
$$= \cos\big((\omega + \omega_c)t\big) + j\ \sin\big((\omega + \omega_c)t\big)$$

Note that this is exactly what happens in the mixer block in our radio model from lecture 2. At the receiving end, the signals are down-converted by mixing with the same carrier i.e.,$\hat{s}(t) = r(t)e^{j\omega_c t}$ and passed through low pass filters to arrive at the recovered I/Q components.

Signal Metrics

- Typical system metrics include size, weight, power and cost (often referred to as SWaP-C)

- Top-level performance metrics like data throughput, or bit-error rate and maximum distance does not capture parameters for sub-system specifications or measurements

  - Important to identify trade-offs and to ensure we do not under- and over-design the building blocks

  - The following parameters can quantify the dynamic performance of the building blocks

Signal Metrics

Property	*Definition*	*MATLAB Function*
SFDR	Spurious free dynamic range	`sfdr`
SINAD	Signal-to-noise-and-distortion ratio	`sinad`
ENOB	Effective number of bits	
SNR	Signal-to-noise ratio	`snr`
THD	Total harmonic distortion	`thd`
THD + N	Total harmonic distortion plus noise	

- SFDR - ratio of rms of signal to rms of worst spurious signal disregarding the bandwidth of interest
- THD - ratio of rms of fundamental to mean of root-sum-square (rss) of harmonics
- THD + N - includes noise components along with harmonics
- SINAD - ratio of rms of signal to mean of rss of all other spectral components - including harmonics, excludes DC
- SNR - ratio of signal to mean of rss of all spectral components, excluding harmonics (i.e., only the noise components)
- ENOB - specifies the actual resolution (or dynamic range) of ADC/DAC to accommodate dynamic quantisation errors in real signals (i.e., finite SNR/SINAD)

Visualising Signal Metrics

Eye Diagram

- Time-domain plots can provide information about changes in signal parameters - including amplitude, frequency, rise time, distortion, noise floor

- Eye diagram is a time-domain display where data signal are constantly sampled and applied to vertical scale, with data rate triggering the horizontal sweep

    - The resulting pattern looks like a series of eyes within the boundaries

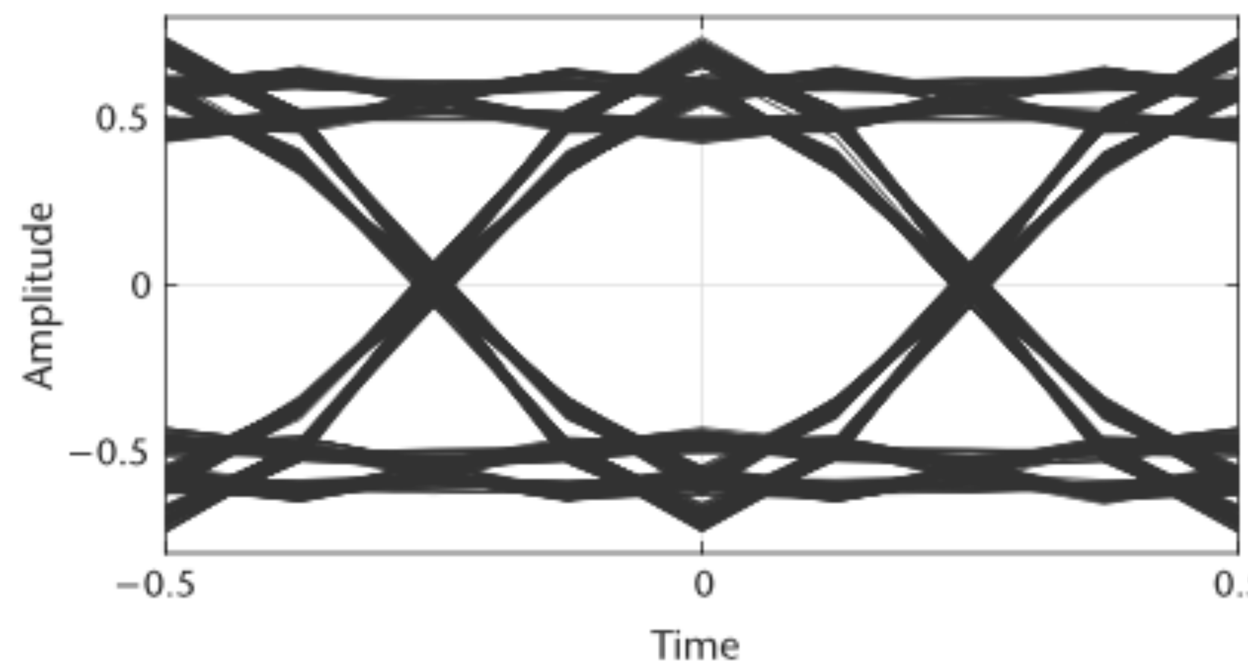
    - The opening of the eye (vertical/horizontal) and the signal lines captures many parameters of the received signal and system

        - Including fast/slow system clock, synchronisation issues, rise/fall times, spurious signals (as overshoot/undershoots)

Visualising Signal Metrics

Eye Diagram

- Two key parameters are width (specifies the time duration during which sampling might be performed) and peak opening time (specifies the optimal time for sampling with minimal noise).
  - Vertical opening - distance between BER threshold point



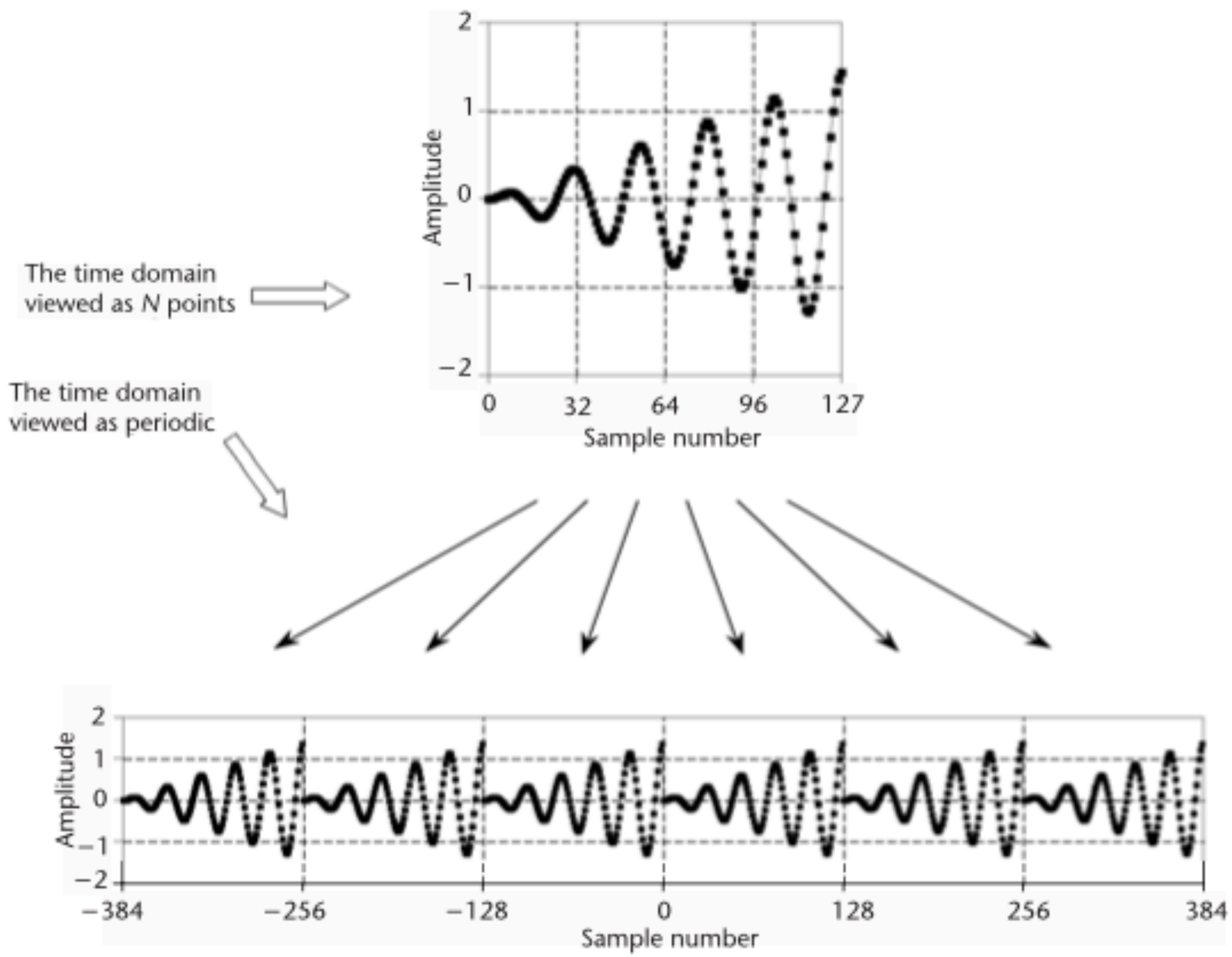
see example in code 2.8 to test it out

Signal processing operations

Time-Frequency Domain & FT

- Time and Frequency domains are alternative ways to represent the same signals - some features captured best in time, while others in frequency
- In most digital systems, we use Discrete Fourier Transform (DFT) through fast Fourier Transform (FFT) and its inverse (IFFT)
  - We mostly use standard FFT kernels or IP blocks - the internal workings are similar to what your learned in 3C1/4C5
- A key difference of DFT from other Fourier Transforms is that both time and frequency domain signals are periodic
  - This is confusing particularly in the time-domain as N-samples taken from the SDR's buffer are usually unrelated

Signal processing operations



This circular nature or periodicity is not real in acquired signals - hence we tend to apply window functions before FFT to remove discontinuities

Signal processing operations

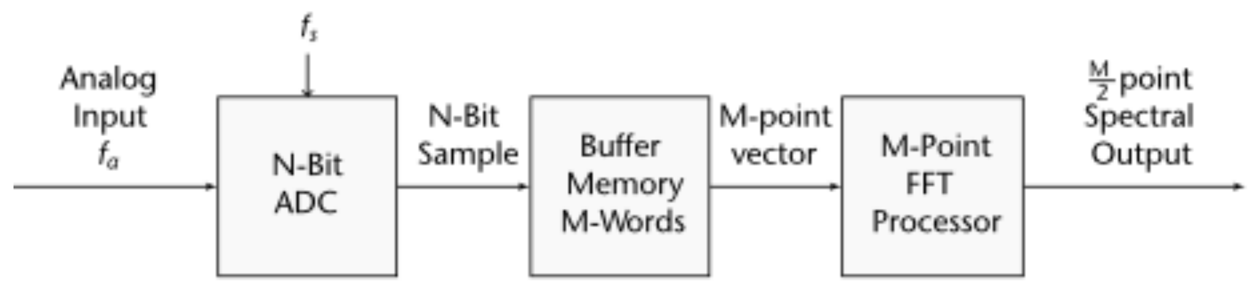
Important properties

Property	*Time Signal*	*Fourier Transform Signal*
Definition	$x(t)$	$\int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$
Inversion formula	$\int_{-\infty}^{\infty} X(\omega)e^{j2\pi\omega t}d\omega$	$X(\omega)$
Linearity	$\sum_{n=1}^{N} a_n x_n(t)$	$\sum_{n=1}^{N} a_n X_n(\omega)$
Symmetry	$x(-t)$	$X(-\omega)$
Time shift	$x(t-t_0)$	$X(\omega)e^{-j\omega t_0}$
Frequency shift	$x(t)e^{j\omega_0 t}$	$X(\omega-\omega_0)$
Scaling	$x(\alpha t)$	$\frac{1}{\vert\alpha\vert}X(\frac{\omega}{\alpha})$
Derivative	$\frac{d^n}{dt^n}x(t)$	$(j\omega)^n X(\omega)$
Integration	$\int_{-\infty}^{\infty} x(\tau)d\tau$	$\frac{X(\omega)}{j\omega}+\pi X(0)\delta(\omega)$
Time convolution	$x(t) * h(t)$	$X(\omega)H(\omega)$
Frequency convolution	$x(t)h(t)$	$\frac{1}{2\pi}X(\omega) * H(\omega)$

* based on [2]. Suppose the time signal is $x(t)$, and its Fourier transform signal is $X(\omega)$

Very similar to z-transform where some operations are simple. see 2.6.3 in the textbook

Signal processing operations



Typical FFT flow involves accumulating M samples of input into a buffer to perform the M-point FFT

- The M is distinct from the N-bit resolution of the digital signal
- The output is a series of $\frac{M}{2}$ points in the frequency domain, spacing between the points being $\frac{f_s}{M}$ (also the resolution or bin width)
- Spectrum covers a total frequency range of DC to $\frac{f_s}{2}$ where f_s is the sampling rate
- Recall that windowing may be applied to eliminate discontinuities

Signal processing operations

Discrete Convolution

- Convolution is the basic building block of digital signal processing
  - Given a signal $x[n]$ with N samples (0 to N-1) and a second signal $h[n]$ with M samples (0 to M-1), then

$$y[i] = h[i] * x[i] = \Sigma_{j=0}^{M-1} h[j] \times x[i-j]$$

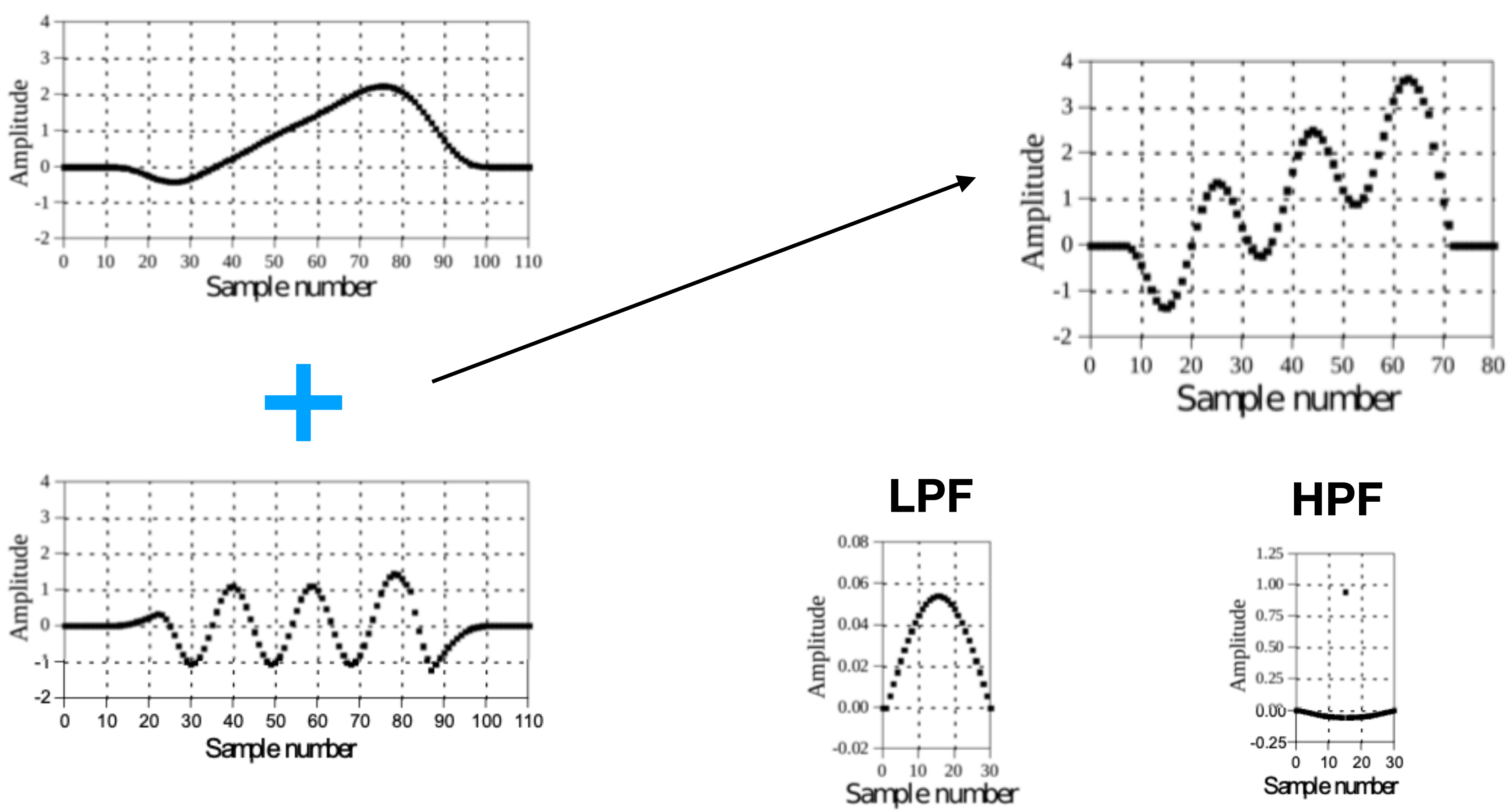
  is called the *convolution sum.* Signal $y[n]$ has N+M-1 samples indexed from 0 to N+M-2

  - **NOTE** that the $*$ operator used for convolution in MATH expressions does not translate to MATLAB or other programming languages
    - MATLAB uses `conv` function to perform convolution
- Essence of convolution - how much each sample of the input signal contributes to many bins of the output signal i.e., how every sample of the input gets transformed to the output signal

Signal processing operations

Discrete Convolution

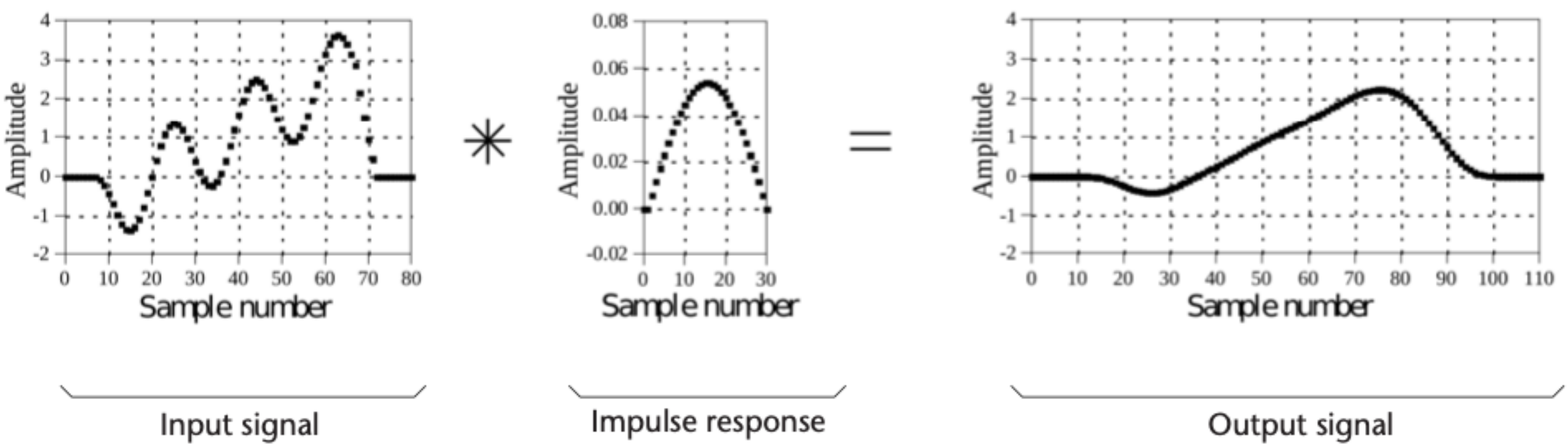
- Convolution describes the behaviour of many building blocks, including filters



Signal processing operations

Discrete Convolution

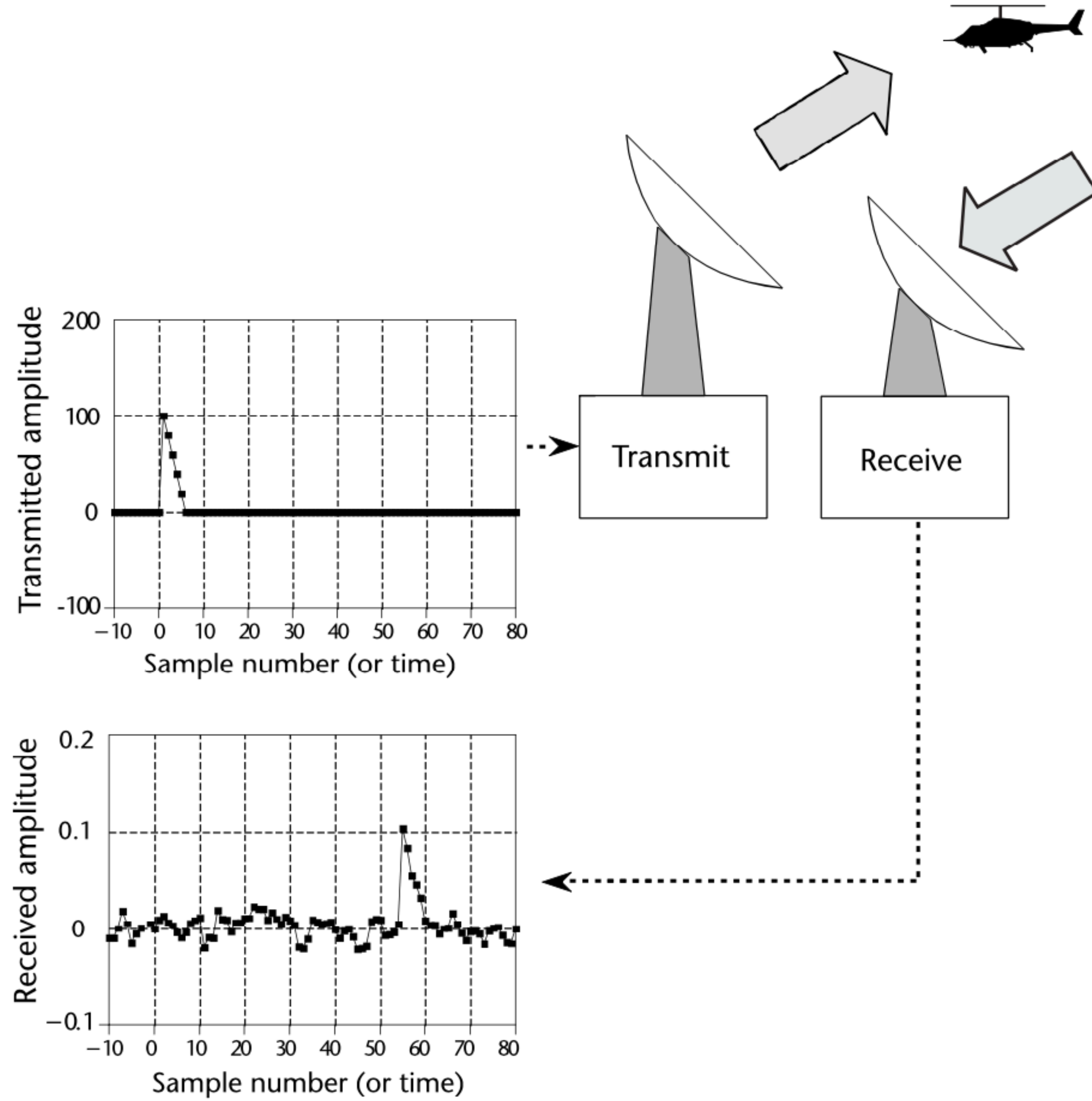
- Convolution describes the behaviour of many building blocks, including filters



Signal processing operations

Correlation

- Cross-correlation measures the similarity of two input sequences as a function of their time-shift wrt each other
- Autocorrelation measures cross-correlation of a signal with itself (and thus, the energy of the signal)
- In the example, to detect the known waveform (the radar sample) from the received noisy waveform, we can use cross-correlation



Signal processing operations

Correlation

- Mathematically, the operation is very similar to convolution; given two signals $f[n]$ and $g[n]$, then cross-correlation between them $(f \star g)[n]$ is given by

$$(f \star g)[n] = \Sigma_{j=-\infty}^{infty} f^*[m] \times g[m+n]$$

where f^* denotes the complex conjugate of f.

- Here, amplitude of each sample in the output is a measure of how much the input signal resembles the known 'target' signal at this specific index (or location in the sample stream)
  - Hence, peak occurs when every received signal resembles the target signal at the specific index - i.e., when the received signal is aligned with the same features as the target signal
  - Also, such a peak will be higher than noise floor - i.e., it can detect a known waveform in a random noisy receiver - this specific type of receiver is called *matched filter receiver*

Note: while the mathematical operation resembles convolution, they are very different signal processing concepts. The similar math however enables algorithmic optimisations for the system

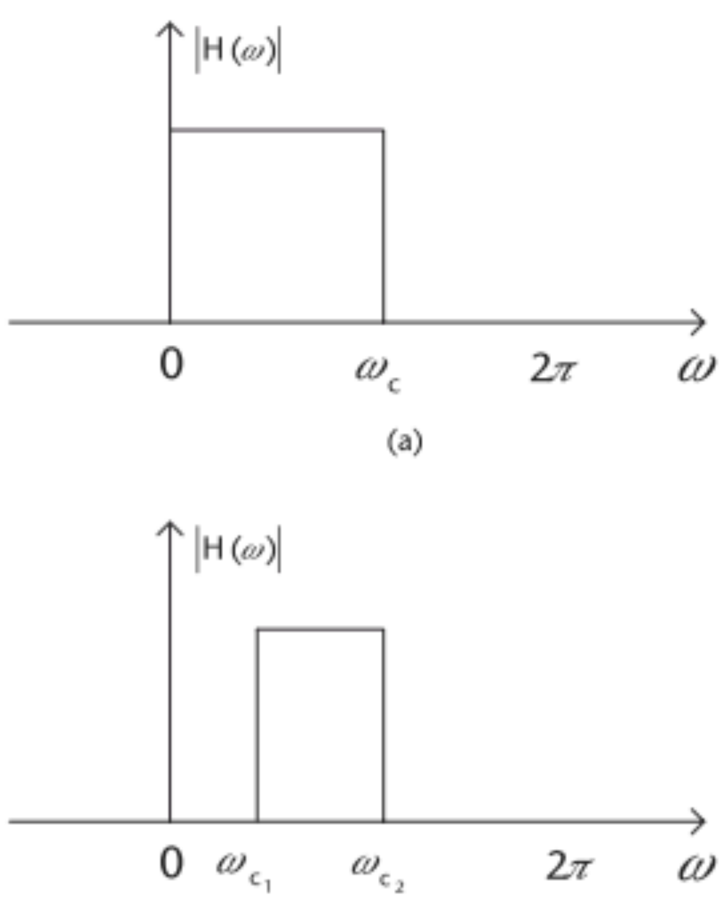
Signal processing operations

Filtering

As we saw before, the convolution operation describes the operation of a filter

- Given an input sequence $x[n]$ and a filter whose impulse response as $h[n]$, the output from the filter block is given by

$$y[n] = x[n] * h[n] \text{ or in frequency domain } Y(\omega) = X(\omega) \times H(\omega)$$



Signal processing operations

Filtering

- Practical digital filters can be described by the difference equation

$$y[n] = -\Sigma_{k=1}^{N} a_k y[n-k] + \Sigma_{k=0}^{M} b_k x[n-k]$$

where $y[n-k]$ are previous filter outputs, $x[n-k]$ are previous and/or current inputs, $\{a_k\}$ are the set of feedback coefficients and $\{b_k\}$ are the set of filter's feed-forward coefficients.

- The filtering problem is thus to determine $\{a_k\}$ and $\{b_k\}$ to approximate the ideal frequency response characteristics required

- A FIR filter will have $\{a_k\} = 0 \, \forall \, k$ resulting in a finite impulse response filter with length M (has only feed-forward and thus only zeros on a pole-zero chart)